

DB Design: database name = "groupdatabase"

"accounts" Table: Forename, Surname, Email{PK}, Password, isDoctor, AssignedDoctor{FK}

"doctors" Table: DoctorID{PK}, Forename, Surname

"messages" Table: msgID{PK}, recipient{FK}, msg, sender

"booking" Table: Date, Time, Patient{FK}, Doctor{FK}, Notes, BookingID{PK}

Sprint 1

(Authentication) -Lead by Rishabh

Whos doing what: Create login page - Rishabh, Login SQL - Luke, Logout Button - Oliver, Landing Page - Rishabh, Getting the message SQL & displayed on the landing page - George

Tests: Make sure a user can only log in with correct details, Check that the system can access and display the user's messages.

Register as new patient (and choose doctor and send confirmation messages) - Lead by Oliver

Whos doing what: Register as a patient (create a new account) - Oliver

Choose doctor from list - Luke Add confirmation message to message tables - George

Tests: Check that confirmation messages are sent, Check that a new patient is successfully registered, Check that a new doctor can be chosen and assigned.

Sprint 2

Enter new booking (and send confirmation messages). - Lead by Oliver

Test to: Check bookings can be added to the database, only if the doctor is available at the given date and time. Check that a confirmation message has been sent.

Reschedule booking (and send confirmation messages). - Lead by George

Test to: see if the patient can have their booking changed, making sure there are no duplicate bookings. And check that confirmation message has been sent

View bookings. - Lead by George

Test: to see if a patient can get a list of their bookings for a given month and year, and make sure the date is valid.

Sprint 3

(Authorisation checks)-Lead by Rishabh

Test: Check if the logs are correct

Change doctor (and send confirmation messages). -Lead by Luke

Tests to: see if the doctor for a user can be changed. Check for confirmation messages.

View visit details and prescriptions. - Lead by Luke

Test to: see if the patient is able to view visit details regarding past bookings, making sure it's for the correct user (the one currently logged in), and not someone else.

Plan for collaboration each week (applies to all sprints):

On Monday at the start of a sprint we will meet and break down the features dividing them up into small bits for each member to do. We will then meet again on Thursday the same week as a quick check up. And then in the second week we will do a mid sprint meeting on Tuesday to see everyone's progress. And on the Thursday of the 2nd week we will meet and finish off all our parts and merge them together on that day or Friday. This will then repeat for each sprint. Junit tests for a method will be made by the person who made the method and checked by the group. The lead for each feature will decide how the work should be split up.

Sprint 2

The system should allow a patient to arrange a booking with their doctor by entering the date and time. If the doctor is not available for the chosen date and time, the system should warn the patient. Otherwise, the system should then send confirmation messages to the patient and the doctor.

- check to see if patient has a doctor - george
- form to create booking (gui) rishabh
- check if doctor is available on date/time & make booking -george
- tests - all create tests for own methods

The system should allow a patient to reschedule a booking with their doctor by entering the date and time. If the doctor is not available for the chosen date and time, the system should warn the patient. Otherwise, the system should then send confirmation messages to the patient and the doctor

- reschedule booking form (gui) - oliver
- check to see if booking is on that date/time - george
- delete booking & make new one - george
- tests -all

The system should allow a patient to view their bookings by entering a month and year

- view patient form (gui) - luke
- get all bookings from selected date - sql put onto form - rishabh
- tests -all

Sprint 3

The system should log all access from a user, i.e. who accessed what functionality and when.

- gui rishabh
- sql -oliver
- test - luke

The system should allow a patient to view the visit details regarding a past booking, for which the doctor provided visit details and prescriptions.

- gui -george
- sql - oliver
- tests - rishabh

–The system should allow a patient to change their doctor using the list of all doctors. The system should then send confirmation messages to the patient and the doctor.

- Reuse implementation in initial choosing of a doctor. - all